PTO/SB/21 (02-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09/823,472
Filing Date	March 30, 2001
First Named Inventor	Thomas E. Wills
Art Unit	2188
Examiner Name	G. Portka
Attorney Docket Number	42390.P8930

Total Number of Pages in This Submission: 66

RECEIVED
CENTRAL FAX CENTER
MAY 11 2004
OFFICIAL

ENCLOSURES *(Check all that apply)*

- [x] Fee Transmittal Form
- [ ] Fee Attached
- [ ] Amendment/Reply
- [ ] After Final
- [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Response to Missing Parts/ Incomplete Application
- [ ] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] After Allowance communication to Technology Center (TC)
- [ ] Appeal Communication to Board of Appeals and Interferences
- [x] Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Other Enclosure(s) (please identify below):

Remarks

Fee Transmittal in duplicate
Appeal Brief under 37 CFR 1.192 in triplicate

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	BLAKELY SOKOLOFF TAYLOR & ZAFMAN Lawrence M. Mennemeier Reg. No. 51,003
Signature	
Date	5-7-2004

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name	Lawrence M. Mennemeier		
Signature		Date	5-7-2004

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/17 (10-03)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 330

Complete if Known	
Application Number	09/823,472
Filing Date	March 30, 2001
First Named Inventor	Thomas E. Willis
Examiner Name	G. Portka
Art Unit	2188
Attorney Docket No.	42390.P8930

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☑ Deposit Account:

Deposit Account Number: 02-2666

Deposit Account Name:

The Director is authorized to: *(check all that apply)*

☑ Charge fee(s) indicated below ☑ Credit any overpayments

☑ Charge any additional fee(s) or any underpayment of fee(s)

☐ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($)

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

		Extra Claims	Fee from below	Fee Paid
Total Claims	-20** =		X	=
Independent Claims	- 3** =		X	=
Multiple Dependent				=

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	330
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 330

SUBMITTED BY				(Complete *(if applicable)*)	
Name (Print/Type)	Lawrence M. Mennemeier	*Registration No. (Attorney/Agent)*	51,003	*Telephone*	408 765-2194
Signature				*Date*	5-7-2004

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

RECEIVED
CENTRAL FAX CENTER
MAY 1 1 2004

OFFICIAL

Docket No.: 42390.P8930

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:
Thomas E. Willis, et al.

Application No. 09/823,472

Filed: March 30, 2001

For: METHOD AND APPARATUS FOR SHARING TLB ENTRIES

Examiner: G. Portka

Art Unit: 2188

CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office, Fax No. (703) 872-9306 on 5-7-04

Date Lawrence M. Mennemeier

APPELLANT'S BRIEF UNDER 37 CFR § 1.192 IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Mail Stop Appeal Brief- Patents
Commissioner of Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant hereby submits this Brief in triplicate in support of an appeal from a final decision of the Examiner, in the above-referenced case. Appellant respectfully requests consideration of this appeal by the board of Patent Appeals and Interference for allowance of the above-referenced patent application.

TABLE OF CONTENTS

I. Real Party in Interest

The real party in interest in the present appeal is Intel Corporation of Santa Clara, California, the assignee of the present application.

II. Related Appeals and Interferences

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.

III. Status of the Claims

Claims 1-30 and 35-46 stand rejected under 35 USC § 102(e) as allegedly being anticipated by US Patent 6,598,050 B1 (Bourekas).

Claims 31-34 are withdrawn from consideration.

IV. Status of Amendments

A first amendment, submitted by appellant on 1/21/2003, in response to a first Office Action mailed 9/19/2002, was entered. In response to appellant's first amendment, a non-final Office Action was mailed 4/3/2003, rejecting the claims on new grounds. Appellant responded in an official response transmitted on 6/12/2003, which was entered. On 8/26/2003, another non-final Office Action was mailed. The 8/26/2003 Office Action was non-final due to the fact that once again new grounds of rejection were introduced. Appellant responded in another official response transmitted on 11/26/2003, which was entered. A Final Office Action was mailed on 12/23/2003. Appellant responded by submitting an amendment and official response after final on 2/4/2004, which was entered and an Advisory Action was mailed 2/17/2004. A Notice of Appeal was transmitted on 3/22/2004, and an appeal ensued.

Accordingly, the claims stand as of the entered amendment of 2/4/2004, and are reproduced in clean form in the Appendix.

V. Summary of the Invention

Appellant's disclosure describes a sharing mechanism for multiple logical processors using a translation lookaside buffer (TLB) to translate virtual addresses into physical addresses. The mechanism supports sharing of TLB entries among logical processors, which may access address spaces in common. The mechanism further supports private TLB entries among logical processors, which may each access a different physical address through identical virtual addresses. Embodiments of the sharing mechanism provide for installation and updating of TLB entries as private entries or as shared entries transparently, without requiring special operating system support or modifications.

In some embodiments, a sharing mechanism in a shared cache may transparently identify if a virtual address translation is sharable between logical processors, and if the virtual address translation is sharable, the sharing mechanism may provide a sharing indication to update a TLB entry, without requiring operating system support or modifications. For example, if a processor initiates a TLB request to look up a virtual address translation and a sharing indication corresponding to the retrieved TLB entry does not include the processor initiating the TLB request, then physical address data and other TLB data may be recovered from page tables in the main memory. Some embodiments may use a mechanism called a page walker to access page tables and compute physical addresses. If a newly constructed virtual address translation matches a retrieved TLB entry, the requesting processor may be transparently added to a set of logical processors sharing the retrieved TLB entry. Thus, the requesting processor may, thereafter, share the TLB entry. Through use of the disclosed sharing mechanism, fast and efficient virtual address translation is provided without requiring expensive functional redundancy or operating system support.

VI. Issues

1. Are claims 1-12, 35 and 40-46 anticipated by Bourekas?
2. Are claims 13-30 anticipated by Bourekas?
3. Are claims 36-39 anticipated by Bourekas?

VII. Grouping of Claims (Independent Claims **Bolded**)

Group I: Operating-system Transparent Methods

For the purposes of this appeal, claims **1**-8, **9**-12, **35** and 40-46 stand or fall together.

Group II: Control Logic to Transparently Produce/Provide a Sharing Indication

For the purposes of this appeal, claims **13**-19 and **20**-30 stand or fall together.

Group III: Storing Dynamically Updated Multithread Sharing Indications

For the purposes of this appeal, claims **36**-39 stand or fall together.

VIII. Argument

A. Claims 1-12 and 35 Are Not Anticipated by Bourekas

Claims 1-12, 35 and 40-46 stand rejected under 35 USC § 102(e) as allegedly being anticipated by US Patent 6,598,050 B1 (Bourekas).

The MPEP § 2131 states that:

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Appellant respectfully submits that in the cited reference, each and every element as set forth in the Group I claims is not found, either expressly or inherently described.

The MPEP § 2111.02 further states that:

> "If the claim preamble, when read in the context of the entire claim, recites limitations of the claim, or, if the claim preamble is 'necessary to give life, meaning, and vitality' to the claim, then the claim preamble should be construed as if in the balance of the claim." Pitney Bowes, Inc. v. Hewlett-Packard Co., 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165-66 (Fed. Cir. 1999).

Claim 1, for example, sets forth:

> 1. (Original) An operating-system transparent method for sharing virtual address translations comprising:
> accessing a virtual address translation; and
> transparently identifying if the virtual address translation is sharable.

The invention of Bourekas relates to a virtual addressing scheme within a microprocessor based system (col. 1, lines 8-9). In this virtual addressing scheme, the virtual addresses have a group membership field (col. 2, lines 27-28). The group membership field is used to permit sharing of data and/or programs among a subset of tasks in a multi-tasking system (col. 2, lines 21-23). With the use of the group membership field, the operating system can support three levels of access in a virtual to physical address translation. The operating system permits a global translation, an

individual translation and a group translation," (col. 5, line 65 through col. 6, line 2, emphasis added).

Therefore, an operating system may be made to use the group membership field of Bourekas to permit three levels of sharing of data and/or programs among tasks. But appellant respectfully submits that such an approach to shared virtual address spaces is not operating system transparent as set forth in claim 1. When operating system support is not provided to permit the shared translations of Bourekas, there is no alternative method either expressly or inherently described to identify or enable shared translations.

The Final Office Action states (emphasis added):

> "Bourekas discloses a method, executable code and processor for transparently sharing virtual address translations, by accessing a translation and identifying if it is sharable...the global bit 115 and group membership fields 118 identify if the translation is sharable, and if so by the current task, as this procedure is transparent to the OS... requires no traps and therefore no modification of or special support by the OS;" and with regard to Claims 35, 40 and 42-46 that, "Bourekas discloses that the TLB stores the indication."

Appellant respectfully disagrees. Appellant submits that when the operating system is not modified to support virtual addresses that have the group membership fields of Bourekas, there is no alternative method to permit shared translations, either expressly or inherently described.

Claim 9 sets forth:

> 9. (Original) A storage medium having executable codes stored thereon for operating-system transparent sharing of virtual address translations which, when executed by a machine, causes the machine to:
> access a virtual address translation; and
> transparently identify if the virtual address translation is sharable.

Claim 35 sets forth:

> 35. (Original) An operating -system transparent method for providing virtual address translations comprising:
> installing an entry in a translation lookaside buffer; and
> transparently enabling sharing of the entry by a plurality of processors.

Claims 1, 9 and 35 clearly set forth operating-system transparent methods that comprise accessing or installing a virtual address translation and transparently identifying if the virtual address translation is sharable or enabling sharing. In contrast, Bourekas describes no operating-system transparent method identifying if the virtual address translation is sharable or enabling sharing, as set forth by the limitations of the Group I claims.

If on one hand, an operating system permits sharing of a virtual address translation in some other way, for example by providing duplicate translation entries in two different page tables, but does not indicate this sharing in the virtual addresses at the location of the group membership fields, Bourekas describes no alternative, operating-system transparent method for identifying if the virtual address translation is sharable or for enabling sharing of a translation lookaside buffer (TLB) entry.

On the other hand, Bourekas describes no operating-system transparent method for identifying when the virtual address translations are actually not sharable. If an operating system provides different and incompatible translation entries in two different page tables, but yet inadvertently uses identical values that correspond to the group membership fields of Bourekas virtual addresses, which ever translation gets installed first for one process may be erroneously identified as shared and then used to mistranslate the virtual addresses of another process. Thus, lack of specific support on the part of the operating system to correctly use the group membership fields of Bourekas could result in erroneous side affects.

Therefore, appellant submits that to conclude that the methods of Bourekas are operating-system transparent would be incorrect. Accordingly in light of the argument

presented above, appellant respectfully submits that independent claims 1, 9 and 35 are not anticipated by Bourekas.

B. Claims 13-19 and 20-30 Are Not Anticipated by Bourekas

Claims 13-19 and 20-30 stand rejected under 35 USC § 102(e) as allegedly being anticipated by US Patent 6,598,050 B1 (Bourekas).

As stated above with regard to the Group I claims, each and every element as set forth in the Group II claims is not found, either expressly or inherently described in the cited reference.

Claim 13 sets forth:

> 13. (Original) A processing system providing operating-system transparent sharing of virtual address translations, the processing system comprising:
> a first logical processor;
> a second logical processor;
> a storage location to store a virtual address translation; and
> a control logic to access a first virtual address translation for the first logical processor in the storage location and to transparently produce a first sharing indication if the virtual address translation may be shared with the second logical processor.

Claim 20 sets forth:

> 20. (Original) An apparatus to provide operating-system transparent sharing of virtual address translations, the apparatus comprising:
> a control logic to access a first virtual address translation for a first processor,
> the control logic further to transparently provide a first sharing indication if the first virtual address translation may be shared with a second processor.

Claims 13 and 20 both set forth apparatus to provide operating-system transparent sharing of virtual address translations comprising control logic to transparently produce or provide a sharing indication if a virtual address translation may be shared.

In contrast, Bourekas describes matching the TLB group membership field and the group membership field in the virtual address (col. 4, lines 64-66). Group membership bits are added to the memory data structures that are used to load the TLB with valid translations (col. 5, lines 29-31). Operating system software conventionally updates the

TLB from a page table of virtual/physical addresses in memory so that a translation can occur (col. 6, lines 50-52).

In other words, the group membership bits in virtual addresses are matched with the group membership bits stored in the TLB from the page tables, both sets of group membership bits being produced and provided by the operating system—not by control logic and not transparent to the operating system. As stated above, if two processes' translations are indeed sharable, but the operating system does not support the group membership bits, the system of Bourekas would not identify the translations as sharable.

Bourekas describes no control logic, either expressly or inherently, to transparently produce or provide a sharing indication to provide operating-system transparent sharing of virtual address translations, as set forth by the limitations of the Group II claims. Accordingly in light of the argument presented above, appellant respectfully submits that independent claims 13 and 20 are not anticipated by Bourekas.

C. Claims 36-46 Are Not Anticipated by Bourekas

Claims 36-39 stand rejected under 35 USC § 102(e) as allegedly being anticipated by US Patent 6,598,050 B1 (Bourekas).

As stated above with regard to the Group I and Group II claims, each and every element as set forth in the Group III claims is not found, either expressly or inherently described in the cited reference.

The MPEP § 2131 also states that:

> "The identical invention must be shown in as complete detail as is contained in the ... claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Claim 36 sets forth:

> 36. (Previously Amended) A multithreading processor comprising:
> an address translation stage including a translation lookaside buffer having a plurality of entries to translate virtual addresses to physical addresses;
> a first entry of the plurality of entries to translate a first virtual address for a first process;
> a control logic comprising circuitry to identify a sharability of the first entry;
> the control logic further to provide a first sharing indication to indicate if the first entry may be shared by a second process; and
> a sharing indication field in the first entry to store the first sharing indication provided by the control logic.

Claim 36 relates to a multithreading processor. Thread processes may include both intra-task processes, which may access address spaces in common, and inter-task processes, which may access different physical address spaces through substantially similar virtual addresses. In general, it cannot be assumed that a TLB match on an entry installed for one thread process can be shared by another thread process. Therefore, even the entries for intra-task processes may need to be identified.

In the multithreading processor of claim 36, a control logic comprises circuitry to identify a sharability of a first TLB entry and to provide a first sharing indication to

indicate if the first entry may be shared by a second process. Claim 36 further sets forth a sharing indication field in the first TLB entry to store the first sharing indication provided by the control logic. Thus, claimed control logic identifies and updates TLB entries as private entries or as shared entries with regard to other thread processes.

In contrast, Bourekas relates to a virtual addressing scheme wherein a group of tasks may be marked for access to a given translation (col. 1, lines 8-9; col. 3, lines 1-3). In Bourekas, both the global bit and the group membership field stored in the TLB entry come from the virtual address translation provided by the operating system (col. 5, line 65 through col. 6, lines 2 and lines 51-52)--not from processor control logic as set forth in claim 36. Matching circuitry of Bourekas receives static group membership fields from the virtual address and from the TLB and simply matches them to determine if they were marked as belonging to the same group (col. 7, line 60 through col. 8, line 2).

Applicant respectfully submits that Bourekas does not expressly or inherently describe, in as complete detail as is set forth by the Group III claims: a multithreading processor with control logic to identify sharability of a TLB entry, to provide a sharing indication to indicate if the first entry may be shared by another process and to store the sharing indication provided by the control logic in a field in the TLB entry. Accordingly in light of the argument presented above, appellant submits that independent claim 36 is not anticipated by Bourekas.

Conclusion

Appellant submits that all claims now pending are in condition for allowance. Such action is earnestly solicited at the earliest possible date. If there is a deficiency in fees, please charge our Deposit Acct. No. 02-2666.

Respectfully submitted,

Date: 5-7-2004

Lawrence M. Mennemeier, Reg. No. 51,003

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8598

IX. Appendix A: Claims Involved in Appeal (Clean Copy)

1. (Original) An operating-system transparent method for sharing virtual address translations comprising:

accessing a virtual address translation; and

transparently identifying if the virtual address translation is sharable.

2. (Original) The method of Claim 1 further comprising

providing a first sharing indication if the virtual address translation is identified as sharable.

3. (Original) The method of Claim 2 further comprising

providing a second sharing indication if the virtual address translation is identified as not sharable.

4. (Original) The method of Claim 2 wherein the first sharing indication indicates a set of logical processes sharing the virtual address translation.

5. (Original) The method of Claim 3 wherein the second sharing indication indicates a private status for the virtual address translation.

6. (Original) The method of Claim 5 wherein the second sharing indication implicitly indicates a private status for the virtual address translation.

7. (Original) The method of Claim 5 wherein the first sharing indication indicates a shared status for the virtual address translation.

8. (Original) The method of Claim 7 wherein the second sharing indication

implicitly indicates a shared status for the virtual address translation.

9. (Original) A storage medium having executable codes stored thereon for operating-system transparent sharing of virtual address translations which, when executed by a machine, causes the machine to:

access a virtual address translation; and

transparently identify if the virtual address translation is sharable.

10. (Original) The storage medium recited in Claim 9 which, when executed by a machine, further causes the machine to:

provide a first sharing indication if the virtual address translation is identified as sharable.

11. (Original) The storage medium recited in Claim 10 wherein the first sharing indication, indicates a set of logical processes sharing the virtual address translation.

12. (Original) The storage medium recited in Claim 10 wherein the first sharing indication, indicates a shared status for the virtual address translation.

13. (Original) A processing system providing operating-system transparent sharing of virtual address translations, the processing system comprising:

a first logical processor;

a second logical processor;

a storage location to store a virtual address translation; and

a control logic to access a first virtual address translation for the first logical processor in the storage location and to transparently produce a first sharing indication if the virtual address translation may be shared with the second logical processor.

14. (Original) The processing system of Claim 13 wherein the first logical processor and the second logical processor are located on the same die.

15. (Original) The processing system of Claim 14 wherein the first logical processor and the second logical processor support multithreading.

16. (Original) The processing system of Claim 13 wherein the virtual address translation is to provide the first and second logical processors access to a shared cache.

17. (Original) The processing system of Claim 16 wherein the sharing indication comprises a set of logical processors sharing the virtual address translation.

18. (Original) The processing system of Claim 16 wherein the shared cache stores data comprising executable instructions for the first and second logical processors.

19. (Original) The processing system of Claim 13 wherein the storage location to store the virtual address translation is further to store the first sharing indication if the virtual address translation may be shared.

20. (Original) An apparatus to provide operating-system transparent sharing of virtual address translations, the apparatus comprising:

a control logic to access a first virtual address translation for a first processor,

the control logic further to transparently provide a first sharing indication if the first virtual address translation may be shared with a second processor.

21. (Original) The apparatus of Claim 20 wherein the control logic comprises:

circuitry to provide a sharing indication to indicate that the virtual address translation translates a virtual address for the second processor.

22. (Original) The apparatus of Claim 20 wherein the control logic comprises:

a combination of circuitry and machine executable processes to compute data of a second virtual address translation for the second processor.

23. (Original) The apparatus of Claim 22 wherein the combination of circuitry and machine executable processes compare the computed data of the second virtual address translation to a corresponding data of the first virtual address translation to identify if the first virtual address translation may be shared with the second processor.

24. (Original) The apparatus of Claim 20 further comprising:

a translation lookaside buffer to store the virtual address translation for the first processor.

25. (Original) The apparatus of Claim 24 wherein the translation lookaside buffer is further to store the first sharing indication if the first virtual address translation may be shared with a second processor.

26. (Original) The apparatus of Claim 24 wherein the first virtual address translation is to provide the first and second processors access to a shared cache.

27. (Original) The apparatus of Claim 26 wherein the shared cache is accessed according to physical addresses.

28. (Original) The apparatus of Claim 20 wherein the control logic is further to provide a second sharing indication if the first virtual address translation may not be shared with the second processor.

29. (Original) The apparatus of Claim 28 wherein the first sharing indication, indicates a set of logical processes sharing the first virtual address translation.

30. (Original) The apparatus of Claim 28 wherein the second sharing indication, indicates a private status for the first virtual address translation.

31-34. (Canceled)

35. (Original) An operating -system transparent method for providing virtual address translations comprising:

installing an entry in a translation lookaside buffer; and

transparently enabling sharing of the entry by a plurality of processors.

36. (Previously Amended) A multithreading processor comprising:

an address translation stage including a translation lookaside buffer having a plurality of entries to translate virtual addresses to physical addresses;

a first entry of the plurality of entries to translate a first virtual address for a first process;

a control logic comprising circuitry to identify a sharability of the first entry;

the control logic further to provide a first sharing indication to indicate if the first entry may be shared by a second process; and

a sharing indication field in the first entry to store the first sharing indication provided by the control logic.

37. (Previously Added) The processor of Claim 36 wherein the sharing indication field indicates a set of processes that have been identified to share the first entry.

38. (Previously Added) The processor of Claim 36 wherein the control logic comprises:

circuitry to provide a sharing indication to indicate that the first entry has an address space identifier data and a virtual address data of the second process.

39. (Previously Added) The processor of Claim 38 wherein the control logic comprises:

a combination of circuitry and state machine executable processes to determine said address space identifier data and said virtual address data of a second virtual address

translation for the second process and to compare them with the first entry.

40. (Previously Added) The method of Claim 35 further comprising

providing a sharing indication to indicate that the entry has been identified as sharable by two or more of said plurality of processors.

41. (Previously Added) The method of Claim 40 wherein the sharing indication indicates a set of logical processors of said plurality of processors that have been identified to share the entry.

42. (Previously Added) The method of Claim 35 further comprising

providing a sharing indication to indicate that the entry has not been identified as sharable by at least one of said plurality of processors.

43. (Previously Added) The method of Claim 42 wherein the sharing indication indicates a private status for the entry.

44. (Previously Added) The method of Claim 43 wherein the sharing indication implicitly indicates a private status for the entry.

45. (Previously Added) The method of Claim 42 wherein the sharing indication indicates a shared status for the entry.

46. (Previously Added) The method of Claim 45 wherein the sharing indication implicitly indicates a shared status for the entry.

RECEIVED
CENTRAL FAX CENTER
MAY 11 2004

OFFICIAL

Docket No.: 42390.P8930

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:
Thomas E. Willis, et al.

Application No. 09/823,472

Filed: March 30, 2001

For: METHOD AND APPARATUS FOR SHARING TLB ENTRIES

Examiner: G. Portka

Art Unit: 2188

CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office, Fax No. (703) 872-9306
on 5-7-04 [signature]
Date Lawrence M. Mennemeier

APPELLANT'S BRIEF UNDER 37 CFR § 1.192 IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Mail Stop Appeal Brief- Patents
Commissioner of Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant hereby submits this Brief in triplicate in support of an appeal from a final decision of the Examiner, in the above-referenced case. Appellant respectfully requests consideration of this appeal by the board of Patent Appeals and Interference for allowance of the above-referenced patent application.

TABLE OF CONTENTS

I. Real Party in Interest

The real party in interest in the present appeal is Intel Corporation of Santa Clara, California, the assignee of the present application.

II. Related Appeals and Interferences

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.

III. Status of the Claims

Claims 1-30 and 35-46 stand rejected under 35 USC § 102(e) as allegedly being anticipated by US Patent 6,598,050 B1 (Bourekas).

Claims 31-34 are withdrawn from consideration.

IV. Status of Amendments

A first amendment, submitted by appellant on 1/21/2003, in response to a first Office Action mailed 9/19/2002, was entered. In response to appellant's first amendment, a non-final Office Action was mailed 4/3/2003, rejecting the claims on new grounds. Appellant responded in an official response transmitted on 6/12/2003, which was entered. On 8/26/2003, another non-final Office Action was mailed. The 8/26/2003 Office Action was non-final due to the fact that once again new grounds of rejection were introduced. Appellant responded in another official response transmitted on 11/26/2003, which was entered. A Final Office Action was mailed on 12/23/2003. Appellant responded by submitting an amendment and official response after final on 2/4/2004, which was entered and an Advisory Action was mailed 2/17/2004. A Notice of Appeal was transmitted on 3/22/2004, and an appeal ensued.

Accordingly, the claims stand as of the entered amendment of 2/4/2004, and are reproduced in clean form in the Appendix.

V. Summary of the Invention

Appellant's disclosure describes a sharing mechanism for multiple logical processors using a translation lookaside buffer (TLB) to translate virtual addresses into physical addresses. The mechanism supports sharing of TLB entries among logical processors, which may access address spaces in common. The mechanism further supports private TLB entries among logical processors, which may each access a different physical address through identical virtual addresses. Embodiments of the sharing mechanism provide for installation and updating of TLB entries as private entries or as shared entries transparently, without requiring special operating system support or modifications.

In some embodiments, a sharing mechanism in a shared cache may transparently identify if a virtual address translation is sharable between logical processors, and if the virtual address translation is sharable, the sharing mechanism may provide a sharing indication to update a TLB entry, without requiring operating system support or modifications. For example, if a processor initiates a TLB request to look up a virtual address translation and a sharing indication corresponding to the retrieved TLB entry does not include the processor initiating the TLB request, then physical address data and other TLB data may be recovered from page tables in the main memory. Some embodiments may use a mechanism called a page walker to access page tables and compute physical addresses. If a newly constructed virtual address translation matches a retrieved TLB entry, the requesting processor may be transparently added to a set of logical processors sharing the retrieved TLB entry. Thus, the requesting processor may, thereafter, share the TLB entry. Through use of the disclosed sharing mechanism, fast and efficient virtual address translation is provided without requiring expensive functional redundancy or operating system support.

VI. Issues

1. Are claims 1-12, 35 and 40-46 anticipated by Bourekas?
2. Are claims 13-30 anticipated by Bourekas?
3. Are claims 36-39 anticipated by Bourekas?

VII. Grouping of Claims (Independent Claims **Bolded**)

Group I: Operating-system Transparent Methods

For the purposes of this appeal, claims **1-8, 9-12, 35** and 40-46 stand or fall together.

Group II: Control Logic to Transparently Produce/Provide a Sharing Indication

For the purposes of this appeal, claims **13-19** and **20-30** stand or fall together.

Group III: Storing Dynamically Updated Multithread Sharing Indications

For the purposes of this appeal, claims **36-39** stand or fall together.

VIII. Argument

A. Claims 1-12 and 35 Are Not Anticipated by Bourekas

Claims 1-12, 35 and 40-46 stand rejected under 35 USC § 102(e) as allegedly being anticipated by US Patent 6,598,050 B1 (Bourekas).

The MPEP § 2131 states that:

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Appellant respectfully submits that in the cited reference, each and every element as set forth in the Group I claims is not found, either expressly or inherently described.

The MPEP § 2111.02 further states that:

> "If the claim preamble, when read in the context of the entire claim, recites limitations of the claim, or, if the claim preamble is 'necessary to give life, meaning, and vitality' to the claim, then the claim preamble should be construed as if in the balance of the claim." Pitney Bowes, Inc. v. Hewlett-Packard Co., 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165-66 (Fed. Cir. 1999).

Claim 1, for example, sets forth:

> 1. (Original) An operating-system transparent method for sharing virtual address translations comprising:
>
> accessing a virtual address translation; and
>
> transparently identifying if the virtual address translation is sharable.

The invention of Bourekas relates to a virtual addressing scheme within a microprocessor based system (col. 1, lines 8-9). In this virtual addressing scheme, the virtual addresses have a group membership field (col. 2, lines 27-28). The group membership field is used to permit sharing of data and/or programs among a subset of tasks in a multi-tasking system (col. 2, lines 21-23). With the use of the group membership field, the operating system can support three levels of access in a virtual to physical address translation. The operating system permits a global translation, an

individual translation and a group translation," (col. 5, line 65 through col. 6, line 2, emphasis added).

Therefore, an operating system may be made to use the group membership field of Bourekas to permit three levels of sharing of data and/or programs among tasks. But appellant respectfully submits that such an approach to shared virtual address spaces is not operating system transparent as set forth in claim 1. When operating system support is not provided to permit the shared translations of Bourekas, there is no alternative method either expressly or inherently described to identify or enable shared translations.

The Final Office Action states (emphasis added):

> "Bourekas discloses a method, executable code and processor for transparently sharing virtual address translations, by accessing a translation and identifying if it is sharable...the global bit 115 and group membership fields 118 identify if the translation is sharable, and if so by the current task, as this procedure is transparent to the OS... requires no traps and therefore no modification of or special support by the OS;" and with regard to Claims 35, 40 and 42-46 that, "Bourekas discloses that the TLB stores the indication."

Appellant respectfully disagrees. Appellant submits that when the operating system is not modified to support virtual addresses that have the group membership fields of Bourekas, there is no alternative method to permit shared translations, either expressly or inherently described.

Claim 9 sets forth:

> 9. (Original) A storage medium having executable codes stored thereon for operating-system transparent sharing of virtual address translations which, when executed by a machine, causes the machine to:
> access a virtual address translation; and
> transparently identify if the virtual address translation is sharable.

Claim 35 sets forth:

> 35. (Original) An operating -system transparent method for providing virtual address translations comprising:
> installing an entry in a translation lookaside buffer; and
> transparently enabling sharing of the entry by a plurality of processors.

Claims 1, 9 and 35 clearly set forth operating-system transparent methods that comprise accessing or installing a virtual address translation and transparently identifying if the virtual address translation is sharable or enabling sharing. In contrast, Bourekas describes no operating-system transparent method identifying if the virtual address translation is sharable or enabling sharing, as set forth by the limitations of the Group I claims.

If on one hand, an operating system permits sharing of a virtual address translation in some other way, for example by providing duplicate translation entries in two different page tables, but does not indicate this sharing in the virtual addresses at the location of the group membership fields, Bourekas describes no alternative, operating-system transparent method for identifying if the virtual address translation is sharable or for enabling sharing of a translation lookaside buffer (TLB) entry.

On the other hand, Bourekas describes no operating-system transparent method for identifying when the virtual address translations are actually not sharable. If an operating system provides different and incompatible translation entries in two different page tables, but yet inadvertently uses identical values that correspond to the group membership fields of Bourekas virtual addresses, which ever translation gets installed first for one process may be erroneously identified as shared and then used to mistranslate the virtual addresses of another process. Thus, lack of specific support on the part of the operating system to correctly use the group membership fields of Bourekas could result in erroneous side affects.

Therefore, appellant submits that to conclude that the methods of Bourekas are operating-system transparent would be incorrect. Accordingly in light of the argument

presented above, appellant respectfully submits that independent claims 1, 9 and 35 are not anticipated by Bourekas.

B. Claims 13-19 and 20-30 Are Not Anticipated by Bourekas

Claims 13-19 and 20-30 stand rejected under 35 USC § 102(e) as allegedly being anticipated by US Patent 6,598,050 B1 (Bourekas).

As stated above with regard to the Group I claims, each and every element as set forth in the Group II claims is not found, either expressly or inherently described in the cited reference.

Claim 13 sets forth:

> 13. (Original) A processing system providing operating-system transparent sharing of virtual address translations, the processing system comprising:
> a first logical processor;
> a second logical processor;
> a storage location to store a virtual address translation; and
> a control logic to access a first virtual address translation for the first logical processor in the storage location and to transparently produce a first sharing indication if the virtual address translation may be shared with the second logical processor.

Claim 20 sets forth:

> 20. (Original) An apparatus to provide operating-system transparent sharing of virtual address translations, the apparatus comprising:
> a control logic to access a first virtual address translation for a first processor,
> the control logic further to transparently provide a first sharing indication if the first virtual address translation may be shared with a second processor.

Claims 13 and 20 both set forth apparatus to provide operating-system transparent sharing of virtual address translations comprising control logic to transparently produce or provide a sharing indication if a virtual address translation may be shared.

In contrast, Bourekas describes matching the TLB group membership field and the group membership field in the virtual address (col. 4, lines 64-66). Group membership bits are added to the memory data structures that are used to load the TLB with valid translations (col. 5, lines 29-31). Operating system software conventionally updates the

TLB from a page table of virtual/physical addresses in memory so that a translation can occur (col. 6, lines 50-52).

In other words, the group membership bits in virtual addresses are matched with the group membership bits stored in the TLB from the page tables, both sets of group membership bits being produced and provided by the operating system–not by control logic and not transparent to the operating system. As stated above, if two processes' translations are indeed sharable, but the operating system does not support the group membership bits, the system of Bourekas would not identify the translations as sharable.

Bourekas describes no control logic, either expressly or inherently, to transparently produce or provide a sharing indication to provide operating-system transparent sharing of virtual address translations, as set forth by the limitations of the Group II claims. Accordingly in light of the argument presented above, appellant respectfully submits that independent claims 13 and 20 are not anticipated by Bourekas.

C. Claims 36-46 Are Not Anticipated by Bourekas

Claims 36-39 stand rejected under 35 USC § 102(e) as allegedly being anticipated by US Patent 6,598,050 B1 (Bourekas).

As stated above with regard to the Group I and Group II claims, each and every element as set forth in the Group III claims is not found, either expressly or inherently described in the cited reference.

The MPEP § 2131 also states that:

> "The identical invention must be shown in as complete detail as is contained in the ... claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Claim 36 sets forth:

> 36. (Previously Amended) A multithreading processor comprising:
> an address translation stage including a translation lookaside buffer having a plurality of entries to translate virtual addresses to physical addresses;
> a first entry of the plurality of entries to translate a first virtual address for a first process;
> a control logic comprising circuitry to identify a sharability of the first entry;
> the control logic further to provide a first sharing indication to indicate if the first entry may be shared by a second process; and
> a sharing indication field in the first entry to store the first sharing indication provided by the control logic.

Claim 36 relates to a multithreading processor. Thread processes may include both intra-task processes, which may access address spaces in common, and inter-task processes, which may access different physical address spaces through substantially similar virtual addresses. In general, it cannot be assumed that a TLB match on an entry installed for one thread process can be shared by another thread process. Therefore, even the entries for intra-task processes may need to be identified.

In the multithreading processor of claim 36, a control logic comprises circuitry to identify a sharability of a first TLB entry and to provide a first sharing indication to

indicate if the first entry may be shared by a second process. Claim 36 further sets forth a sharing indication field in the first TLB entry to store the first sharing indication provided by the control logic. Thus, claimed control logic identifies and updates TLB entries as private entries or as shared entries with regard to other thread processes.

In contrast, Bourekas relates to a virtual addressing scheme wherein a group of tasks may be marked for access to a given translation (col. 1, lines 8-9; col. 3, lines 1-3). In Bourekas, both the global bit and the group membership field stored in the TLB entry come from the virtual address translation provided by the operating system (col. 5, line 65 through col. 6, lines 2 and lines 51-52)--not from processor control logic as set forth in claim 36. Matching circuitry of Bourekas receives static group membership fields from the virtual address and from the TLB and simply matches them to determine if they were marked as belonging to the same group (col. 7, line 60 through col. 8, line 2).

Applicant respectfully submits that Bourekas does not expressly or inherently describe, in as complete detail as is set forth by the Group III claims: a multithreading processor with control logic to identify sharability of a TLB entry, to provide a sharing indication to indicate if the first entry may be shared by another process and to store the sharing indication provided by the control logic in a field in the TLB entry. Accordingly in light of the argument presented above, appellant submits that independent claim 36 is not anticipated by Bourekas.

Conclusion

Appellant submits that all claims now pending are in condition for allowance. Such action is earnestly solicited at the earliest possible date. If there is a deficiency in fees, please charge our Deposit Acct. No. 02-2666.

Respectfully submitted,

Date: 5-7-2004

Lawrence M. Mennemeier, Reg. No. 51,003

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8598

IX. Appendix A: Claims Involved in Appeal (Clean Copy)

1. (Original) An operating-system transparent method for sharing virtual address translations comprising:

accessing a virtual address translation; and

transparently identifying if the virtual address translation is sharable.

2. (Original) The method of Claim 1 further comprising

providing a first sharing indication if the virtual address translation is identified as sharable.

3. (Original) The method of Claim 2 further comprising

providing a second sharing indication if the virtual address translation is identified as not sharable.

4. (Original) The method of Claim 2 wherein the first sharing indication indicates a set of logical processes sharing the virtual address translation.

5. (Original) The method of Claim 3 wherein the second sharing indication indicates a private status for the virtual address translation.

6. (Original) The method of Claim 5 wherein the second sharing indication implicitly indicates a private status for the virtual address translation.

7. (Original) The method of Claim 5 wherein the first sharing indication indicates a shared status for the virtual address translation.

8. (Original) The method of Claim 7 wherein the second sharing indication

implicitly indicates a shared status for the virtual address translation.

9. (Original) A storage medium having executable codes stored thereon for operating-system transparent sharing of virtual address translations which, when executed by a machine, causes the machine to:

access a virtual address translation; and

transparently identify if the virtual address translation is sharable.

10. (Original) The storage medium recited in Claim 9 which, when executed by a machine, further causes the machine to:

provide a first sharing indication if the virtual address translation is identified as sharable.

11. (Original) The storage medium recited in Claim 10 wherein the first sharing indication, indicates a set of logical processes sharing the virtual address translation.

12. (Original) The storage medium recited in Claim 10 wherein the first sharing indication, indicates a shared status for the virtual address translation.

13. (Original) A processing system providing operating-system transparent sharing of virtual address translations, the processing system comprising:

a first logical processor;

a second logical processor;

a storage location to store a virtual address translation; and

a control logic to access a first virtual address translation for the first logical processor in the storage location and to transparently produce a first sharing indication if the virtual address translation may be shared with the second logical processor.

14. (Original) The processing system of Claim 13 wherein the first logical processor and the second logical processor are located on the same die.

15. (Original) The processing system of Claim 14 wherein the first logical processor and the second logical processor support multithreading.

16. (Original) The processing system of Claim 13 wherein the virtual address translation is to provide the first and second logical processors access to a shared cache.

17. (Original) The processing system of Claim 16 wherein the sharing indication comprises a set of logical processors sharing the virtual address translation.

18. (Original) The processing system of Claim 16 wherein the shared cache stores data comprising executable instructions for the first and second logical processors.

19. (Original) The processing system of Claim 13 wherein the storage location to store the virtual address translation is further to store the first sharing indication if the virtual address translation may be shared.

20. (Original) An apparatus to provide operating-system transparent sharing of virtual address translations, the apparatus comprising:

a control logic to access a first virtual address translation for a first processor,

the control logic further to transparently provide a first sharing indication if the first virtual address translation may be shared with a second processor.

21. (Original) The apparatus of Claim 20 wherein the control logic comprises:

circuitry to provide a sharing indication to indicate that the virtual address translation translates a virtual address for the second processor.

22. (Original) The apparatus of Claim 20 wherein the control logic comprises:

a combination of circuitry and machine executable processes to compute data of a second virtual address translation for the second processor.

23. (Original) The apparatus of Claim 22 wherein the combination of circuitry and machine executable processes compare the computed data of the second virtual address translation to a corresponding data of the first virtual address translation to identify if the first virtual address translation may be shared with the second processor.

24. (Original) The apparatus of Claim 20 further comprising:

a translation lookaside buffer to store the virtual address translation for the first processor.

25. (Original) The apparatus of Claim 24 wherein the translation lookaside buffer is further to store the first sharing indication if the first virtual address translation may be shared with a second processor.

26. (Original) The apparatus of Claim 24 wherein the first virtual address translation is to provide the first and second processors access to a shared cache.

27. (Original) The apparatus of Claim 26 wherein the shared cache is accessed according to physical addresses.

28. (Original) The apparatus of Claim 20 wherein the control logic is further to provide a second sharing indication if the first virtual address translation may not be shared with the second processor.

29. (Original) The apparatus of Claim 28 wherein the first sharing indication, indicates a set of logical processes sharing the first virtual address translation.

30. (Original) The apparatus of Claim 28 wherein the second sharing indication, indicates a private status for the first virtual address translation.

31-34. (Canceled)

35. (Original) An operating -system transparent method for providing virtual address translations comprising:

installing an entry in a translation lookaside buffer; and

transparently enabling sharing of the entry by a plurality of processors.

36. (Previously Amended) A multithreading processor comprising:

an address translation stage including a translation lookaside buffer having a plurality of entries to translate virtual addresses to physical addresses;

a first entry of the plurality of entries to translate a first virtual address for a first process;

a control logic comprising circuitry to identify a sharability of the first entry;

the control logic further to provide a first sharing indication to indicate if the first entry may be shared by a second process; and

a sharing indication field in the first entry to store the first sharing indication provided by the control logic.

37. (Previously Added) The processor of Claim 36 wherein the sharing indication field indicates a set of processes that have been identified to share the first entry.

38. (Previously Added) The processor of Claim 36 wherein the control logic comprises:

circuitry to provide a sharing indication to indicate that the first entry has an address space identifier data and a virtual address data of the second process.

39. (Previously Added) The processor of Claim 38 wherein the control logic comprises:

a combination of circuitry and state machine executable processes to determine said address space identifier data and said virtual address data of a second virtual address

translation for the second process and to compare them with the first entry.

40. (Previously Added) The method of Claim 35 further comprising

providing a sharing indication to indicate that the entry has been identified as sharable by two or more of said plurality of processors.

41. (Previously Added) The method of Claim 40 wherein the sharing indication indicates a set of logical processors of said plurality of processors that have been identified to share the entry.

42. (Previously Added) The method of Claim 35 further comprising

providing a sharing indication to indicate that the entry has not been identified as sharable by at least one of said plurality of processors.

43. (Previously Added) The method of Claim 42 wherein the sharing indication indicates a private status for the entry.

44. (Previously Added) The method of Claim 43 wherein the sharing indication implicitly indicates a private status for the entry.

45. (Previously Added) The method of Claim 42 wherein the sharing indication indicates a shared status for the entry.

46. (Previously Added) The method of Claim 45 wherein the sharing indication implicitly indicates a shared status for the entry.